



# Photo Studio

Autumn 2025



# Problem & Scope

- Lack of centralization
- Time spent
- Lack of instant logging
- Creation of orders
- Transmission of orders between the roles.
- Price calculation (express)
- Track of consumable materials
- Generation of daily reports

# User stories

- As the receptionist, I want to create orders with accurate information so that I don't give out wrong orders.
- As a receptionist I want to create two types of orders so the clients and photographers can have their own order.
- As the receptionist, I want to automatically calculate daily income so that I don't report wrong numbers.
- As the photographer, I want to have the orders automatically so I can complete orders as soon as possible.
- As the photographer, I want to have an automatic system to calculate daily resource usage so that I don't have to keep track of it and less mistakes happen.

# CRC cards

## Client

Responsibilities	Collaborators
Place an order (photo printing or film developing)	Receptionist, Order

## Receptionist

Responsibilities	Collaborators
Records the clients data Passes orders to the photographer and clients Collect payment upon completion* Generate daily revenue report	Client, Order, Photographer, Administrator

## Administrator

Responsibilities	Collaborators
Review reports from receptionist and materials	Photographer, Receptionist

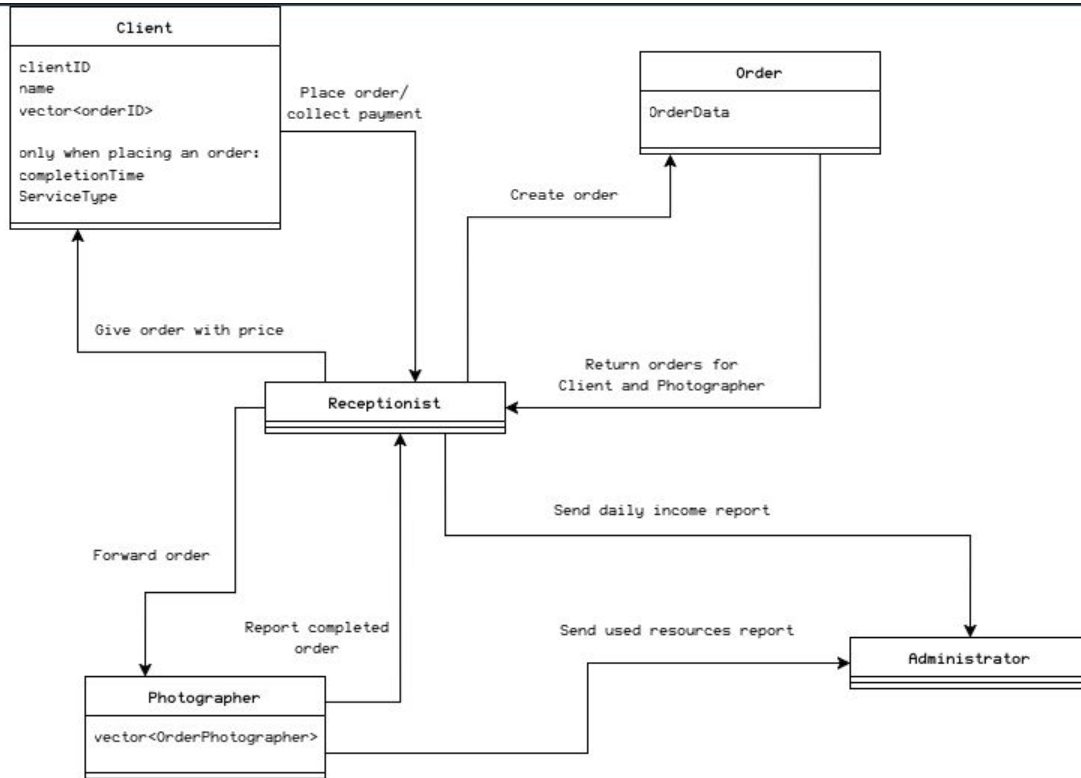
## Order

Responsibilities	Collaborators
Create client type order (Client, type, completion time, urgency) Create a photographer type order (Num, type, completion time and urgency) Calculate price (surcharge if express)	Client, Receptionist, Photographer

## Photographer

Responsibilities	Collaborators
Processes order and return results Use consumables materials Submits report on consumed materials	Receptionist, Order, Administrator

# Diagram



# Traceability table

User Story	CRC Responsibility	Simple Class Diagram Element	C++ Element
Receptionist creates the order	<b>Receptionist:</b> Records the client data <b>Order:</b> Calculates price	Receptionist -- Client - - Order	<code>Receptionist::createOrder()</code> <code>Order::calculatePrice()</code>
Order generates two child orders	<b>Order:</b> Create client type order Create photographer type order	Order -- Client -- Photographer	<code>Order::generateOrders()</code>
Photographer process the orders	<b>Photographer:</b> Processes order and return results Use consumable materials	Photographer -- Order	<code>Photographer::processOrder()</code>
Receptionist collects the payment	<b>Client:</b> Pay for completed order Receptionist: Collect payment upon completion	Receptionist -- Client - - Order	<code>Receptionist::collectPayment()</code>
Receptionist generates a daily revenue report	<b>Receptionist:</b> Generates daily revenue report	Receptionist -- Administrator	<code>Receptionist::generateReport()</code>
Photographer submits a report on consumed materials	<b>Photographer:</b> Submit report on consumed materials	Photographer -- Administrator	<code>Photographer::submitReport</code>